Getting State-Space Models from FEM Simulations

A.W.M. (Jos) van Schijndel
Eindhoven University of Technology
P.O. Box 513; 5600 MB Eindhoven; Netherlands, A.W.M.v.Schijndel@tue.nl

Abstract:. Finite element based modeling is one of the most powerful computational tools currently available. Amongst others a possible drawback could be the computation duration time, that can be expected at transient nonlinear problems, or at more simple problems with a large time span. One of the possible solutions is trying simplify the FEM model into a lower order system without losing its characteristic behavior. In this paper we study the possibilities of state-space models as candidate lower order systems of FEM models using three different approaches: (1) Using the state-space model export function of Comsol itself; (2) Using a system identification tool to retrieve a black box state-space model; (3) using inverse modeling to get a lumped parameter state-space model.
Our methodology was to start with a simple benchmark in Comsol, compare the three above mentioned approaches and proceed by adding more complexity into the next benchmark. Thus obtaining results for different benchmarks.
We conclude all approaches are capable of significantly reduce computation duration time without loss of accuracy. Comparing the three approaches from a physical point of view, the grey-box model is preferable because its parameters (state-space matrices) have a physical meaning and therefore parameters studies can be done without the necessity to simulate the FEM model over and over again. Finally, the reader should notice that no general conclusions can be obtained from this rather limited study.

Keywords: FEM, State-space, Computation, Speed.

1. Introduction

Finite element based modeling is one of the most powerful computational tools currently available. Amongst others a possible drawback could be the computation duration time, that can be expected at transient nonlinear problems, or at more simple problems with a large time span. One of the possible solutions is trying simplify the FEM model into a lower order system without losing its characteristic behavior as visualized in Figure 1.

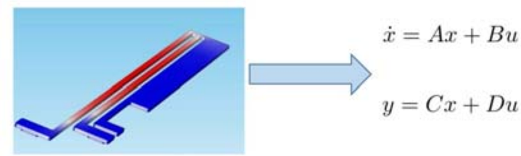


Figure 1. The approach of getting State-Space models from FEM Simulations

The aim of this work is to study the possibilities of State-Space (SS) systems to significantly reduce the simulation duration time without loss of accuracy. Section 2 presents an existing reference case from the building energy simulation community. Section 3 shows the state-space models as candidate lower order systems of FEM models using three different approaches. (1) Using the state-space model export function of Comsol itself; (2) Using a system identification tool to retrieve a black box state-space model; (3) using inverse modeling to get a lumped parameter state-space model. Section 4 provides the conclusions.

2. The reference case

A very suitable reference case was found at the current International Energy Agency Annex 58 (2012). It concerns a test box with overall dimension 120x120x120 cm³. Floor, roof and three of the four walls are opaque, one wall contains a window with opening frame. Details of the overall geometry with the exact dimensions can be found in figure 2.

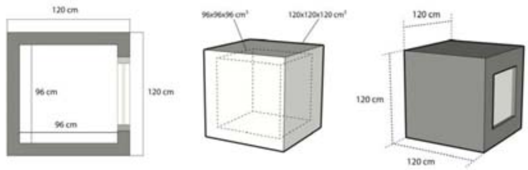
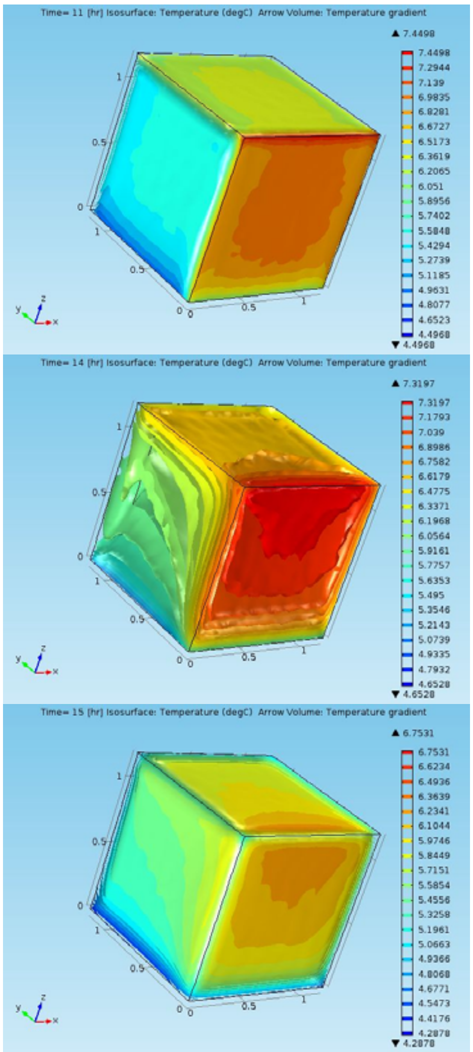


Figure 2. The reference case.

We started to build a 3D model of the opaque test box, heavy weight, air change rate: ACH=0 using Comsol. In order to compare the Comsol 3D FEM model with the HAMBase (HAMLab

2017) lumped model, an equivalent heat conduction of the air is used in Comsol instead of CFD. The distribution in the test box is simulated using Dutch weather data. Figure 4 shows the 3D dynamics snapshots of the isosurfaces. The main challenge now is how to match the high resolution distributed temperature results of Comsol with the lumped temperature results of the BES model. For this reference case (opaque test box, heavy weight, ACH=0) we were able to get a very good match by using a so-called equivalent heat conduction coefficient for the air inside the box in Comsol. Figure 3 presents snapshots of the simulated temperature distribution during the day.



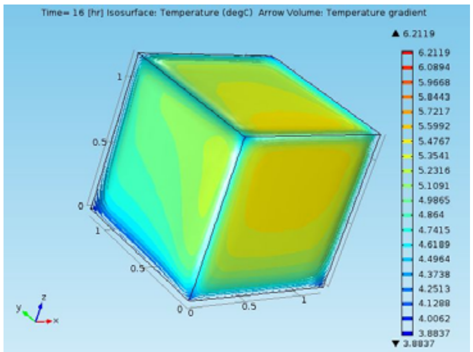


Figure 3 3D dynamics snapshots of the temperature isosurfaces.

Figure 4 shows the comparison of the simulated mean indoor air temperature using Comsol (blue line) and HAMBase (green line) during the first month. The verification result is very satisfactory.

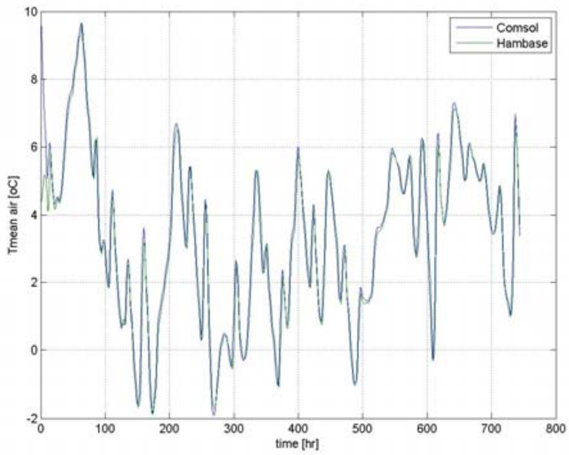


Figure 4. Comparison of the simulated mean indoor air temperature using Comsol (blue line) and HAMBase (green line) during the first month.

3. Using State-Space Systems to reduce simulation duration

A State-Space (SS) system is a system that consists of set of linear differential equations, with state vector x, an input vector u and an output vector y. See Figure 5.

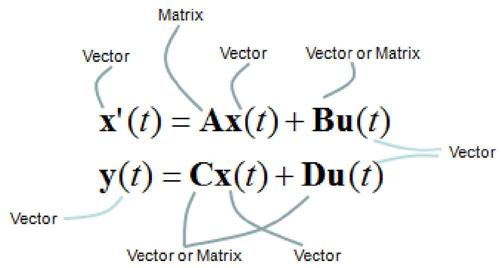


Figure 5. A state space system representation.

There are several practical ways to apply State-Space (SS) systems for reducing the simulation duration time:

(1) Creating a full state-space matrix system using specific COMSOL functionality and including reduced order systems
(2) Using identification techniques for example the MatLab identification Toolbox to fit SS systems
(3) Creating a lumped parameter SS model from first principles, where parameters have a physical meaning.

Each SS application is presented in a separate Section.

3.1 Full and reduced SS

We used the Comsol mphstate.m function in MatLab to retrieve (sparse) matrices, A (6395 x 6395), B (6395 x 1), C (1 x 6395), D (=0) and x0 (6395 x 1) for the FEM model. Using MatLab (commands ss , lsim) this gives, as expected, exact the same results as the Comsol FEM results. However due to large number of matrix elements the computation time duration reduction is quite limited. One possibility is to use reduced order techniques to decrease the size of the above mentioned matrices. We use the Comsol-MatLab interface for extracting the full state space model from the Comsol solution and to reduce the order to 8. The MatLab code is show below:

```
%Extract full SS model
M2 =
mphstate(model,'sol1','out',{'A'
'B' 'C' 'D' 'x0'},...
  'input','mod1.var1', 'output',
'mod1.dom1');
```

```
%Create system in MatLab
sys2= ss(M2.A,M2.B,M2.C,M2.D);

%Simulate full SS
y2=lsim(sys2,u,t,M2.x0);

%Reduce order
Options = balredOptions();
sys2Reduced2 =
balred(sys2,8,Options);

%Simulate reduced SS
y3=lsim(sys2Reduced2,u,t);
```

The results are presented in Figures 6 and 7:

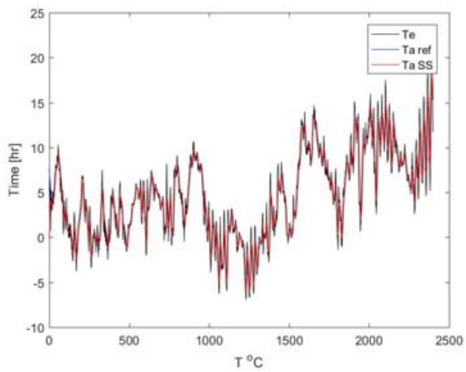


Figure 6 Validation using the reduced 8th order state space model

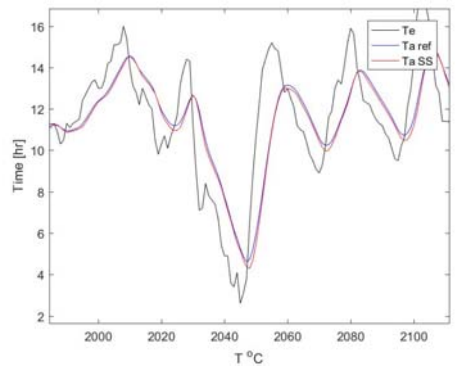


Figure 7 Validation using the reduced 8th order state space model, zoomed in

The 8th order SS model performs very well.

3.2 SS from identification techniques

For identification techniques two types of sets are required: A training data set and a validation data set. Each data set should include time series for at least one input and one output. In our case we have:

Training data set (see Figures 8 and 9).
Input: One year (1981) hourly based external air temperature (Te)
Output: One year hourly based simulated mean indoor air temperature (Ta ref).

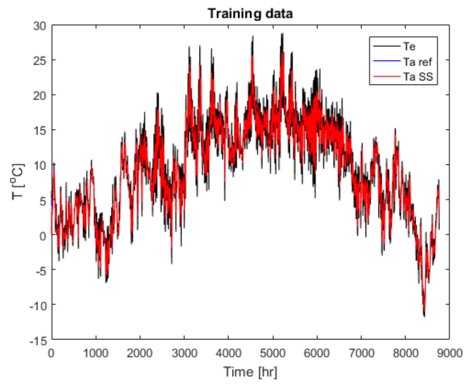


Figure 8 Training data set

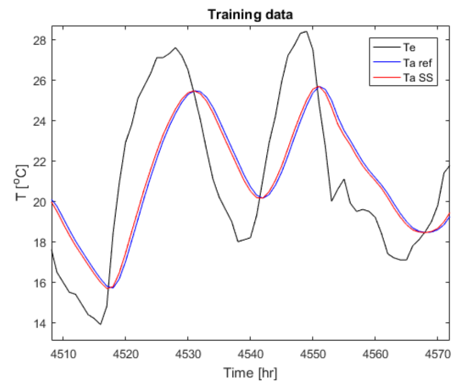


Figure 9 Training data set, zoomed in

We used the MatLab system identification Toolbox to retrieve a 99.4% fit for the following 4 order SS system:

A=

2.9240e-04	-3.2314e-04	6.0115e-04	-8.4876e-06
0.0019	-0.0019	0.0034	-3.4333e-05
-2.2344e-04	2.2694e-04	-3.9655e-04	-1.3116e-04
0.0042	-0.0042	0.0081	-4.1391e-05

B=

-5.3540e-04
-0.0030
-3.1878e-04
-0.0064

C=

-126.2529	37.3281	-0.9173	-6.7066

D=0;

The results of this model is already included in figures 8-11 (label Ta SS). Below the validation results are presented.

Validation data set (see Figures 10 and 11).
Input: One year (1982) hourly based external air temperature (Te)
Output: One year hourly based simulated mean indoor air temperature (Ta ref).

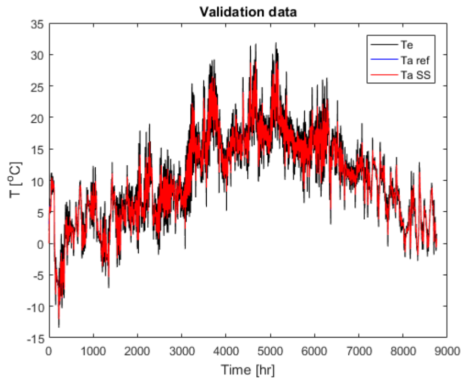


Figure 10 Validation data set

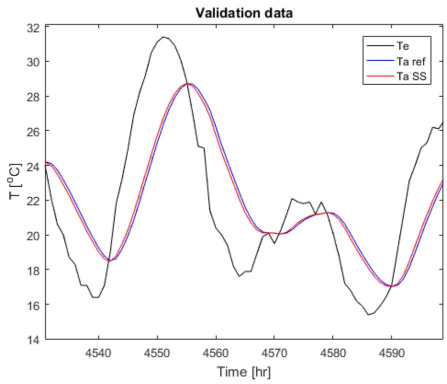


Figure 11 Validation data set, zoomed in

The results of this model is already included in figures 8-11 (label Ta SS). It can be seen from the validation data results (figures 10 and 11) that the above mentioned SS system is strikingly capable to capture the FEM based dynamics and therefore an almost perfect replacement for the more complex FEM model.

3.3 SS from lumped parameter modeling

In this Section we develop a SS model from lumped parameter modeling techniques. In Figure 12, a corresponding heat transfer network with the FEM model is shown:

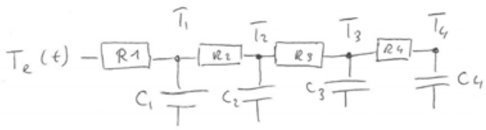


Figure 12. The heat transfer network model

Where R is heat resistance [K/W], C is heat capacity [J/K]. T1 is the external surface temperature, T2 is the inner construction temperature, T3 is the internal surface temperature, T4 is the indoor air temperature and Te is the external air temperature. The ordinary differential equations (Odes) that can be derived from Figure 12 are presented in Figure 13.

$$C_1 \frac{dT_1}{dt} = \frac{T_e - T_1}{R_1} - \frac{T_1 - T_2}{R_2}$$

$$C_2 \frac{dT_2}{dt} = \frac{T_1 - T_2}{R_2} - \frac{T_2 - T_3}{R_3}$$

$$C_3 \frac{dT_3}{dt} = \frac{T_2 - T_3}{R_3} - \frac{T_3 - T_4}{R_4}$$

$$C_4 \frac{dT_4}{dt} = \frac{T_3 - T_4}{R_4}$$

Figure 13. The accompanying ODEs of the heat transfer network model.

The above ODEs can be written as a state-space representation. By taking state vector x =[T1; T2; T3; T4], output vector y=T4 and input vector u=Te. The procedure is shown in Figure 14.

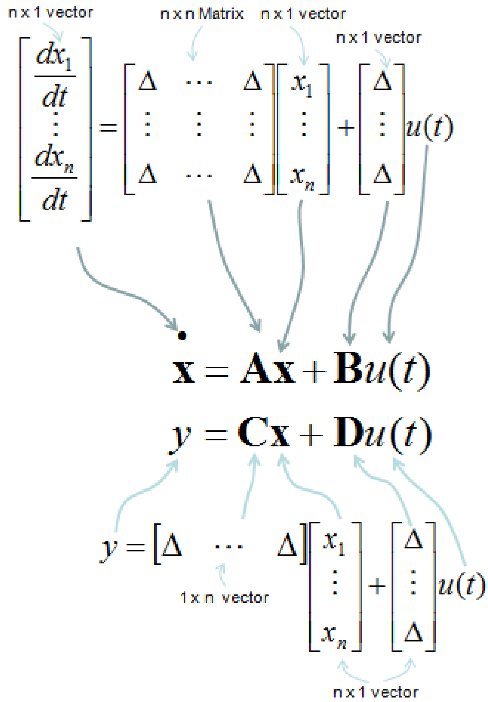


Figure 14. Relation between the ODEs and the state-space matrices A,B,C,D

We calculated A,B,C,D using MatLab, the program code is shown below:

```
hi=1/0.13; %internal surf coef
he=25;  %external surf coef
A=5*1.2*1.2; %total surface
d1=0.06; %thickness material
k1=2.9; %heat cond. Coef.
c1=1000; %spec. heat
rho1=2750;% density
%Layer 2 = Layer 1
d2=d1;
k2=k1;
c2=c1;
rho2=rho1;

ca=1000;%spec. heat air
rhoa=1.2; %density air
Va=0.96*0.96; %air volume
%R calc
R1=1/(he*A);
R2=d1/(k1*A);
R3=d2/(k2*A);
R4=1/(hi*A);
%C calc
C1=0.5*c1*rho1*d1*A;
C3=0.5*c2*rho2*d2*A;
C2=C1+C3;
C4=ca*rhoa*Va;

%A calc
A(1,1)=(-1/(R1*C1)-1/(R2*C1));
A(1,2)=(           1/(R2*C1));
A(2,1)=(1/(R2*C2));
A(2,2)=(-1/(R2*C2)-1/(R3*C2));
A(2,3)=(           1/(R3*C2));
A(3,2)=(1/(R3*C3));
A(3,3)=(-1/(R3*C3)-1/(R4*C3));
A(3,4)=(           1/(R4*C3));
A(4,3)=(1/(R4*C4));
A(4,4)=(-1/(R4*C4))
%B calc
B(1,1)=(1/(R1*C1));
B(2,1)=0;
B(3,1)=0;
B(4,1)=0;
%C calc
C=[0 0 0 1];
%D calc
D=0;
```

This gives the following numerical results for the SS matrices.

A=

-8.8889e-04	5.8586e-04	0	0
2.9293e-04	-5.8586e-04	2.9293e-04	0
0	5.8586e-04	-6.7910e-04	9.3240e-05
0	0	0.0501	-0.0501

B=

3.0303e-04
0
0
0

C=[0 0 0 1];

D=0;

Because this is a so-called forward model we don't need training data. Again we use MatLab (commands ss, lsim) to simulate the validation data. The results are shown in Figure 15 and 16

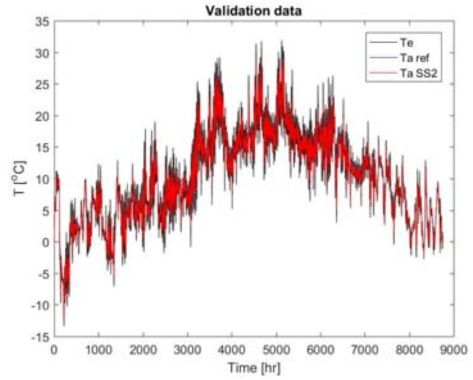


Figure 15. Validation

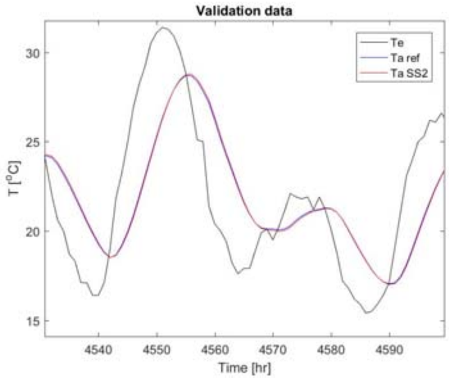


Figure 16. Validation zoomed

Again we see a very good SS model for representing a FEM simulation.

4. Conclusions

We conclude that all approaches: are capable of significantly reduce computation duration time without loss of accuracy. Comparing the three approaches from a physical point of view, the lumped parameter model is preferable because its parameters (state-space matrices) have a physical meaning and therefore parameters studies can be done without the necessity to simulate the FEM model over and over again. Finally, the reader should notice that no general conclusions can be obtained from this rather limited study.

References

Gontikaki, M. & A.W.M. van Schijndel (2009) Application of system identification methods to implement Comsol models into external simulation environments, Proceedings of the COMSOL Conference 2009 Milan

HAMLab (2017), http://archbps1.campus.tue.nl/bpswiki/index.php/Hamlab

International Energy Agency Annex 58 (2017). http://www.ecbcs.org/annexes/annex58.htm

Schijndel, A.W.M. van & Kramer, R.P. (2013). The benefits of FEM-SS-BES (Finite Element Method, State-Space, Building Energy Simulation) modeling exchange for building physics. Proceedings of the 2nd central European conference on building physics, 9-11 September 2013, Vienna, Austria. (pp. 253-260). Vienna: Vienna University of Technology